Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently Amended) A method comprising:

storing in memory a plurality of queue descriptors each including a head pointer pointing to a first element in a corresponding queue and a tail pointer pointing to a last element in the corresponding queue;

in response to a command to perform an enqueue or dequeue operation with respect to ~~the~~ a first queue, fetching from the memory to a cache one of either the head pointer or tail pointer of a first queue descriptor corresponding to the first queue; and

returning to the memory from the cache portions of the first queue descriptor modified by the operation.

2. (Currently Amended) The method of claim 1 including fetching the head pointer and not the tail pointer of the first queue descriptor in response to a command to perform a dequeue operation with respect to the first queue.

3. (Currently Amended) The method of claim 1 including fetching the tail pointer and not the head pointer of the first queue descriptor in response to a command to perform an enqueue operation with respect to the first queue.

4. (Currently Amended) The method of claim 1 including returning to memory the head pointer and not the tail pointer of the first queue descriptor if only dequeue operations were performed on the first queue.

5. (Currently Amended) The method of claim 1 including returning to memory the tail pointer and not the head pointer of the first queue descriptor if only enqueue operations were performed on the first queue while the first queue was unempty.

6. (Currently Amended) The method of claim 1 including returning to memory the head pointer and tail pointer of the first queue descriptor if an enqueue and a dequeue operation were performed on the first queue, or an enqueue operation was performed on the first queue while the first queue was empty.

7. (Currently Amended) A method comprising:

storing in memory a plurality of queue descriptors each including a head pointer pointing to a first element in a corresponding queue and a tail pointer pointing to a last element in the corresponding queue;

determining whether a head pointer or a tail pointer of a queue descriptor that was fetched from the memory to a cache in response to an enqueue or dequeue operation had been modified by the enqueue or dequeue operation; and

returning one of either the head pointer or tail pointer to the memory from the cache only if that pointer had been modified.

8. (Original) The method of claim 7 including using valid bits in the cache to track modifications to the pointers.

9. (Original) The method of claim 8 including using a first valid bit to track modifications to the head pointer and second valid bit to track modifications to the tail pointer.

10. (Original) The method of claim 9 including setting the first valid bit if a dequeue operation is performed with respect to the queue descriptor, or an enqueue operation is performed with respect to the queue descriptor while the queue is empty.

11. (Original) The method of claim 9 including setting the second valid bit if an enqueue operation is performed with respect to the queue descriptor.

12. (Original) The method of claim 9 including setting a pointer's valid bit when the pointer is fetched from the memory to the cache.

13. (Original) The method of claim 9 including returning to the memory pointers whose valid bits have been set.

14. (Currently Amended) An apparatus comprising:

memory for storing a plurality of queue descriptors, each of which includes a head pointer pointing to a first element in a corresponding queue and a tail pointer pointing to a last element in the corresponding queue;

a cache for storing queue descriptors corresponding to up to a number of the memory's queue descriptors; and

a processor configured to:

fetch from the memory to the cache one of either the head pointer or the tail pointer of a particular queue descriptor in response to a command to perform an enqueue or a dequeue operation with respect to the particular queue descriptor; and

return to the memory from the cache portions of the queue descriptor modified by the operation.

15. (Original) The apparatus of claim 14 wherein the processor is configured to fetch the head pointer and not the tail pointer in response to a command to perform a dequeue operation.

16. (Original) The apparatus of claim 14 wherein the processor is configured to fetch the tail pointer and not the head pointer in response to a command to perform an enqueue operation.

17. (Original) The apparatus of claim 14 wherein the processor is configured to return to the memory the head pointer and not the tail pointer when only dequeue operations were performed on the queue.

18. (Original) The apparatus of claim 14 wherein the processor is configured to return to the memory the tail pointer and not the head pointer if only enqueue operations were performed on the queue while the queue was unempty.

19. (Original) The apparatus of claim 14 wherein the processor is configured to return to the memory the head pointer and tail pointer if an enqueue and a dequeue operation were performed on the queue, or an enqueue operation was performed on the queue while the queue was empty.

20. (Previously Presented) The apparatus of claim 14 wherein the cache stores valid bits and wherein the processor is configured to track modifications to the pointers in the cache by setting the valid bits.

21. (Original) The apparatus of claim 20 wherein the cache stores a first valid bit to track modifications to the head pointer and a second valid bit to track modifications to the tail pointer.

22. (Original) The apparatus of claim 21 wherein the processor is configured to set the first valid bit if a dequeue operation is performed with respect to the queue descriptor, or an enqueue operation is performed with respect to the queue descriptor while the queue is empty.

23. (Original) The apparatus of claim 21 wherein the processor is configured to set the second valid bit if an enqueue operation is performed with respect to the queue descriptor.

24. (Original) The apparatus of claim 21 wherein the processor is configured to set a pointer's valid bit when the pointer is fetched from the memory to the cache.

25. (Original) The apparatus of claim 21 wherein the processor is configured to return to the memory the pointers whose valid bits has been set.

26. (Currently Amended) An article comprising a computer-readable medium that stores computer-executable instructions for causing a computer system to:

store in memory a plurality of queue descriptors each including a head pointer pointing to a first element in a corresponding queue and a tail pointer pointing to a last element in the corresponding queue;

in response to a command to perform an enqueue or dequeue operation with respect to a first queue corresponding to a first queue descriptor, fetch from memory to a cache one of either a head pointer of the first queue descriptor pointing to a first element in ~~a~~ the first queue or a tail pointer of the first queue descriptor pointing to a last element in the first queue; and

return to the memory from the cache the portions of the first queue descriptor modified by the operation.

27. (Currently Amended) The article of claim 26 including instructions to cause the computer system to:

fetch the head pointer and not the tail pointer of the first queue descriptor in response to a command to perform a dequeue operation with respect to the first queue; or

fetch the tail pointer and not the head pointer of the first queue descriptor in response to a command to perform an enqueue operation with respect to the first queue.

28. (Currently Amended) The article of claim 26 including instructions to cause the computer system to return to memory:

the head pointer and not the tail pointer of the first queue descriptor if only dequeue operations are performed on the first queue;

the tail pointer and not the head pointer of the first queue descriptor if only enqueue operations are performed on the first queue while the first queue is unempty; or

both the head pointer and tail pointer of the first queue descriptor if both an enqueue and a dequeue are performed on the first queue, or an enqueue operation was performed on the first queue while the first queue is empty.

29. (Original) The article of claim 26 including instructions to cause the computer system to set a valid bit corresponding to a pointer in the cache when the pointer is modified by an enqueue or a dequeue operation.

30. (Original) The article of claim 29 including instructions to cause the computer system to return to the memory pointers whose corresponding valid bits are set.

31. (New) A method comprising:

storing in memory a queue descriptor for a queue, the queue descriptor including a count identifying a number of elements in the queue, a head pointer pointing to a first element in the queue and a tail pointer pointing to a last element in the queue;

in response to a command to perform an enqueue or dequeue operation with respect to the queue, fetching from the memory to a cache the count and one of either the head pointer or tail pointer; and

returning to the memory from the cache portions of the queue descriptor modified by the operation.

32. (New) The method of claim 31 including fetching the count and the head pointer and not the tail pointer in response to a command to perform a dequeue operation; or fetching the count and the tail pointer and not the head pointer in response to a command to perform an enqueue operation.

33. (New) A method comprising:

determining whether a head pointer or a tail pointer of a queue descriptor that was fetched from memory to a cache in response to an enqueue or dequeue operation on a queue had been modified by the enqueue or dequeue operation; and

returning a count identifying a number of elements in the queue and one of either the head pointer or tail pointer to the memory from the cache only if that pointer had been modified.

34. (New) An apparatus comprising:

memory for storing queue descriptors which include a count identifying a number of elements in a queue, a head pointer pointing to a first element in the queue and a tail pointer pointing to a last element in the queue;

a cache for storing queue descriptors corresponding to up to a number of the memory's queue descriptors; and

a processor configured to:

fetch from the memory to the cache the count and one of either the head pointer or the tail pointer of a particular queue descriptor in response to a command to perform an enqueue or a dequeue operation with respect to the particular queue descriptor; and

return to the memory from the cache portions of the queue descriptor modified by the operation.

35. (New) The apparatus of claim 34 wherein the processor is configured to fetch the count and the head pointer and not the tail pointer in response to a command to perform a dequeue operation; or fetch the count and the tail pointer and not the head pointer in response to a command to perform an enqueue operation.

36. (New) An article comprising a computer-readable medium that stores computer-executable instructions for causing a computer system to:

store in memory a queue descriptor including a count identifying a number of elements in a queue, a head pointer pointing to a first element in the queue and a tail pointer pointing to a last element in the queue;

in response to a command to perform an enqueue or dequeue operation with respect to a queue, fetch from memory to a cache the count and one of either a head pointer pointing to a first element in the queue or a tail pointer pointing to a last element in the queue; and

return to the memory from the cache the portions of the queue descriptor modified by the operation.

37. (New) The article of claim 36 including instructions to cause the computer system to:

fetch the count and the head pointer and not the tail pointer in response to a command to perform a dequeue operation; or

fetch the count and the tail pointer and not the head pointer in response to a command to perform an enqueue operation.